Cluster Based Vector Attribute Filtering

Kiwanuka, Fred N.; Wilkinson, Michael H.F.

Published in:
Mathematical Morphology - Theory and Applications

DOI:
10.1515/mathm-2016-0007

Publication date:
2016

Research Article **Open Access**

Fred N. Kiwanuka and Michael H.F. Wilkinson

Cluster Based Vector Attribute Filtering

DOI 10.1515/mathm-2016-0007
Received June 29, 2015; accepted February 8, 2016

Abstract: Morphological attribute filters operate on images based on properties or attributes of connected components. Until recently, attribute filtering was based on a single global threshold on a scalar property to remove or retain objects. A single threshold struggles in case no single property or attribute value has a suitable, usually multi-modal, distribution. Vector-attribute filtering allows better description of characteristic features for 2D images. In this paper, we apply vector-attribute filtering to 3D and incorporate unsupervised pattern recognition, where connected components are classified based on the similarity of feature vectors. Using a single attribute allows multi-thresholding for attribute filters where more than two classes of structures of interest can be selected. In vector-attribute filters automatic clustering avoids the need for either setting very many attribute thresholds, or finding suitable class prototypes in 3D and setting a dissimilarity threshold. Explorative visualization reduces to visualizing and selecting relevant clusters. We show that the performance of these new filters is better than those of regular attribute filters in enhancement of objects in medical images.

Keywords: Image enhancement, Object detection, Attribute Filters, Connected Operators, Max Tree, Clustering

1 Introduction

The field of connected mathematical morphology has contributed a wide range of operators to image processing. Efficient techniques and algorithms have been developed for extracting image components that are useful in the representation and description of shapes. These techniques have found application in medical imaging [8, 18, 38], document analysis [20], video processing [24], content-based compression [33], color processing [32] and remote sensing applications [42].

In many applications, an important task is to extract particular regions of an image while preserving as much of the contour information as possible. This is the main aim of connected filters [25, 27], a strictly edge preserving class of operators in mathematical morphology. These operators act by merging flat zones given some criteria, and filter an image without introducing new contours.

An important sub-class of connected filters are attribute filters [2, 24]. They allow filtering based on the properties or *attributes* of connected components in the image. Examples of attribute filters include attribute openings, closings, thickenings, and thinnings[2, 24, 38].

Attribute openings [2, 24] allow the use of size based attributes. By contrast, attribute thinnings allow use of shape-based attributes, which require translation, scale and rotation invariant descriptors.

Despite the development of many types of attributes, in their current format, attribute filters have two drawbacks. First, the attributes used are often a single scalar value describing either size or shape properties of connected components. This works well if the desired structures can be separated easily from undesired structures, especially when attributes of high discriminative power can be found [7, 8, 19, 38]. However, in

Fred N. Kiwanuka: College of Computing and Information Sciences, Makerere University, P.O.Box 7062 Kampala, Uganda, E-mail: kiwanoah@gmail.com
Michael H.F. Wilkinson: Institute for Mathematics and Computing Science, University of Groningen, P.O Box 407, 9700 AK Groningen, Netherlands, E-mail: m.h.f.wilkinson@rug.nl

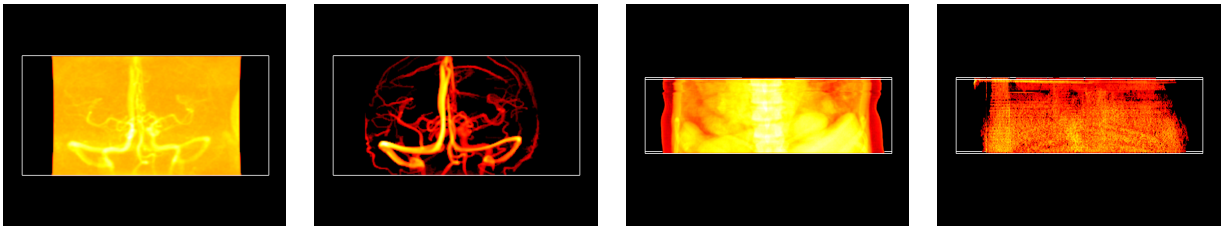

Figure 1: Attribute Filtering:(a)original:*angiolarge* (b)filtered with non-comp(λ=3.2) (c)original:*kidney-stone* (d)filtered with non-comp(λ=7.8)

many cases, objects of different classes are not easily discriminated by a single shape number. An example of this is shown in Fig. 1. The *non-compactness* is a known robust attribute filter and easily performs well on relatively low-clutter volumes, such as the (angiolarge) data set in Fig. 1(b). However, on a more noisy image with more clutter, like kidney-stone, the filter completely fails, as shown in Fig. 1(d). This could imply that perhaps a single threshold or attribute is deficient.

This is the reason why vector-attribute filtering [16, 34, 36] was proposed. This allows a better description of characteristic features, such as size and shape of the objects in the image. These features have been studied for synthetic images of characters and 2D dermatological images, and are based on dissimilarity measures such as Euclidean or Mahalanobis distance in feature space. Components that are similar to a set of reference shapes can be preserved or removed. This has been applied when a priori knowledge of a suitable reference shape is known. In many image filtering tasks, such as medical images, a priori knowledge of a given object is not readily available. In this research we develop 3D vector-attribute filters which do not rely on reference shapes.

Secondly, in attribute filtering, segmentation is achieved by grouping all those components with attributes greater than a threshold into one class, and all other components into another class. Global thresholding works well when the attribute properties are relatively uniform and the regions of interest in the image differ significantly from the background features. Ideally, the distribution of the attribute values of all components should be bimodal. Thresholding performs poorly if there are more than 2 peaks in the attribute distribution, causing any fixed threshold for the entire image to fail in separating the objects from the background.

In this research, we propose improving the robustness and the versatility of attribute filters by developing vector-attribute filters in which features are selected or rejected based on feature vectors, just as in [16, 34, 36], rather than a single property. Unlike the previous work, we apply this to 3D medical volume data sets, in which the selection of reference shapes is far more difficult than in the 2D case, where we can simply delineate features of interest on the screen. Therefore, we want to develop a method of interactive attribute filtering which does not need prior knowledge of ideal target shapes, and preferably requires minimal a-priori setting of parameters. To achieve this we adapt unsupervised pattern recognition approaches, where object classes are learned based on the clustering of attribute vectors. We demonstrate the capacity of these approaches on 3D biomedical images for both size and shape based attributes.

The paper is organized as follows. The theory of vector-attribute filters is covered in Section 3. In Section 4, a description of attribute cluster filter computation and implementation for vector-attribute filtering is built. Section 5, presents experimental results obtained for the vector-attributes in 3D medical image enhancement, where a comparison to other methods is presented. A discussion of results is also in this section. We give concluding remarks in Section 6.

2 Theory

This section briefly reviews the concept of connected attribute filters, and then describes the theory of vector-attribute filtering. In the following, E is some non-empty, universal set and $\mathcal{P}(E)$the family of all subsets of E. Binary images X, Y are subsets of E, and grey-scale images are mapping from E to some subset of $\mathbb{R}$ or $\mathbb{Z}$. A set $C \subseteq X$ is a *connected component* of X if C is connected, and if for any *connected* set D, $C \subseteq D \subseteq X$ implies $C = D$. In other words C is a maximal connected subset of X.

2.1 Attribute Filters

An operator ψ [25, 27] operating on binary image X is connected if and only if the set difference $X \backslash \psi(X)$ is exclusively composed of the connected components of X or its complement X^c. This means the filters act solely by merging connected components. To access these connected components, we utilize *connectivity openings* $\Gamma_x(X)$, which are families of operators, indexed by $x \in E$, that extract the union of all connected sets within X that have a point $x \in E$ in their intersection. This union is the maximal connected set $C \subseteq X$, such that $x \in C$, i.e. it is the connected component containing x [23, 28].

Definition 1. *A binary connectivity opening $\Gamma_x(X)$ of X at a point $x \in E$ with $\mathcal{C} \subseteq \mathcal{P}(E)$ some connectivity class, can be defined as*

$$\Gamma_x(X) = \begin{cases} \bigcup \{C_i \in \mathcal{C} | x \in C_i, C_i \subseteq X\} & \text{if } x \in X \\ \emptyset & \text{otherwise.} \end{cases} \tag{1}$$

A connectivity class $\mathcal{C} \subseteq \mathcal{P}(E)$ is the family of all connected sets in E [28]. A connectivity opening is an algebraic opening, which means it is *anti-extensive*, i.e., $\Gamma_x(X) \subseteq X$, *increasing* or *isotone*, i.e., $X \subseteq Y \Rightarrow \Gamma_x(X) \subseteq \Gamma_x(Y)$ and *idempotent* i.e., $\Gamma_x(\Gamma_x(X)) = \Gamma_x(X)$. Furthermore, for all, $X \subseteq E$, $x, y \in E$, $\Gamma_x(X)$ and $\Gamma_y(X)$ are equal or disjoint: if $\Gamma_x(X) \cap \Gamma_y(X) \neq \emptyset \Rightarrow \Gamma_x(X) = \Gamma_y(X)$.

To remove or retain these connected components is the role of *attribute filters*. Attribute filters [2, 24] are defined based on a family of connectivity openings, by imposing constraints on the connected components they return. Such constraints are expressed in the form of binary criteria which decide to accept or to reject components based on some attribute measure or criterion. Breen and Jones [2] note that the attribute criterion usually has the form

$$\Lambda(C) \equiv Attr(C) \geq \lambda, \tag{2}$$

with $Attr(C)$ some real valued attribute of C and λ the attribute threshold. They do note that other options are equally valid.

After extracting the connected components using connectivity openings, a trivial filter ψ^Λ, based on attribute criterion Λ is applied to each. These are defined as

$$\psi^\Lambda(C) = \begin{cases} C & \text{if } \Lambda(C) \text{ is true} \\ \emptyset & \text{otherwise} \end{cases} \tag{3}$$

The complete attribute filter ψ_Λ based on criterion Λ can be defined as

$$\psi_\Lambda(X) = \bigcup_{x \in X} \psi^\Lambda(\Gamma_x(X)) \tag{4}$$

This is the union of all connected *foreground* components which meet criterion Λ. This operator is an attribute opening only if Λ is increasing. Otherwise it is an attribute thinning. The dual counterparts of these operators are attribute closings and thickening, respectively, and can be defined as

$$\Psi_\Lambda(X) = (\psi_\Lambda(X^c))^c \tag{5}$$

where $X^c = E \setminus X$ denotes the complement of X. This removes connected *background* components which do not meet the criterion Λ. As noted above, Λ usually takes the form of comparing some property, such as surface area, volume or any other property, to a predefined threshold, and returning true if the value is above the threshold and false otherwise.

Gray-level connected operators [2, 24, 26] rely on the notion of partitions of flat zones. A partition is a set of non overlapping, non void regions that fills the entire space, in our case the image domain E. Connected operators for gray scale images work on connected components of level set images L_h of image f given by

$$L_h(f) = \{x \in E | f(x) = h\}. \tag{6}$$

for gray level h from some set G. For convenience we tend to drop the image f, and just refer to $L_h(f)$ as L_h. At every level L_h is a binary image where connected components and connectivity openings can be carried out. The connected components L_h^k at level h with k the index of connected component (peak)are the so called *flat zone*. If E is the image domain then

$$E = \bigcup_{h \in G} \bigcup_{k \in I} L_h^k, \tag{7}$$

and

$$L_h^k \cap L_j^m = \emptyset. \tag{8}$$

This applies for all $(k, h) \neq (m, j)$. A grey-scale operator ψ [26] is connected if given an input image f the partition(P_f) of E of flat zones of f is finer than the partition of flat zones of its output($P_{\psi(f)}$) after transformation due to the connected operator ψ. We denote this as

$$P_f \preccurlyeq P_{\psi(f)} \tag{9}$$

in which the partial order $\preccurlyeq$ is defined as

$$A \preccurlyeq B \quad \equiv \quad \forall a \in A,\ \exists b \in B : a \subseteq b. \tag{10}$$

An extensive discussion of operators working on hierarchies of partitions is given by Soille [31].

The simplest way to extend these increasing filters to gray scale is through threshold decomposition [15] and attribute filters. The principle works by thresholding the image at all possible levels, then applying the filter to each level, and finally stacking the results. The threshold set T_h at level h can be defined as:

$$T_h(f) = \{x \in E | f(x) \geq h\}. \tag{11}$$

Using these threshold sets, the gray-scale attribute filter ψ_Λ can be computed from its binary counterpart Ψ_Λ using the expression

$$\psi_\Lambda(f)(x) = \sup\{h | x \in \Psi_\Lambda(T_h(f))\}. \tag{12}$$

More complicated extensions are used in the case of non-increasing filters, as discussed in [2, 24, 35].

3 Vector-Attribute Filtering

In the binary case, attribute filters [2], retain those connected components of an image, which meet certain criteria. After computing the connected components, some property or attribute of each component is computed. A threshold is usually applied to these attributes to determine which components are retained and which removed. Thus, the criterion Λ, usually has the form

$$\Lambda(C) = Attr(C) \geq \lambda \tag{13}$$

with C the connected component, $Attr(C)$ some real-valued attribute of C and λ the attribute threshold. For grey scale image f, we compute these attributes for the connected components of threshold sets $X_h(f)$, defined as

$$X_h(f) = \{x \in E | f(x) \geq h\}. \tag{14}$$

Urbach et al. [34] replaced the single attribute by a feature vector of dimensionality D. Rather than setting D thresholds, they based the criterion on dissimilarity to a reference vector $\vec{r}$, ideally obtained from some reference shape.

They define a multi-variate attribute thinning $\Phi^{\{\Lambda_i\}}(X)$ with scalar attributes τ_i and their corresponding criteria $\{\Lambda_i\}$, with $1 \leq i \leq D$, such that connected components are preserved if they satisfy at least one of the criteria $\{\Lambda_i(C)\}$ and are removed otherwise:

$$\Phi^{\{\Lambda_i\}}(X) = \bigcup_{i=1}^{D} \Phi^{\Lambda_i}(X) \quad \text{with} \quad \Lambda_i(C) \equiv \tau_i(C) \geq \lambda_i. \tag{15}$$

where λ_i are the attribute thresholds.

The set of scalar attributes τ_i can also be considered as a single vector-attribute $\vec{\tau} = \{\tau_1, \tau_2, \ldots, \tau_D\}$, in which case a vector-attribute thinning is needed with a criterion:

$$\Lambda^{\vec{\tau}}_{\vec{\lambda}} \equiv \exists i : \tau_i(C) \geq \lambda_i \quad \text{for } 1 \leq i \leq D. \tag{16}$$

with $\vec{\lambda}$ the attribute threshold vector. Urbach et al. [34] then proceed to a more useful criterion defined as

$$\Lambda^{\vec{\tau}}_{\vec{r},\epsilon}(C) \equiv d(\vec{\tau}(C), \vec{r}) \geq \epsilon \tag{17}$$

where d is some dissimilarity measure, $\vec{r}$ is a reference vector, and ϵ is a dissimilarity threshold. This replaces D parameters with just a single value ϵ, but adds the need for a reference vector. A binary vector-attribute thinning $\Phi^{\vec{\tau}}_{\vec{r},\epsilon}(X)$, with D-dimensional vectors, removes the connected components of a binary image X, whose vector-attributes differ more than a given quantity from a reference vector $\vec{r} \in Y$.

Alternatively [16] suggested

$$\Lambda^{\vec{\tau}}_{\vec{r},\epsilon}(C) \equiv d(\vec{\tau}(C), \vec{r}) \leq \epsilon \tag{18}$$

which is essentially the complement (but not quite) of the form of [34]. Therefore, this preserves all objects with attribute vectors sufficiently similar to the reference, rejecting all others. We will work with this latter form in the remainder of the paper.

While it is possible to compose reference shapes in the 2D case of letters from a known font [34], in 3D it becomes much harder. Therefore, it could be useful to consider approaches that do not need these reference vectors a priori.

The approach here derives some inspiration from [36]. This paper introduces the notion of *context attributes* of components. Context attributes describe how a component relates to other components in the image. Alignment, distance, and similarities in size, shape, and orientation between the individual components are used to determine which components belong to the same object. Contextual filter preserves only those components which visually appear to belong to a certain group of similar components. Urbach [36] only considers the binary case, and focuses mainly on spatial relations such as proximity and alignment. Here we move to grey scale and volume data, and focus exclusively on similarities in terms of (vector-)attributes.

Another related approach is that of Xu et al. [43], who propose a method for filtering Max-Trees with non-increasing attribute, by building a Max-Tree of a Max-Tree. The non-increasing attribute takes on the role of grey level, and the parent-child relationship takes on the role of the neighborhood relationship in building this secondary Max-Tree. Filtering this secondary Max-Tree can select for zones of similar attributes (or extrema), and as such could be interpreted as clustering nodes based on combined attributes and parent-child relations. This method cannot deal with vector attributes, however. We will therefore not perform a comparison. A similar approach in [21], was published after submission of our initial work [9], and such a comparison would be of interest in future work. It proposes a hierarchical Markovian unsupervised algorithm in order to classify the nodes of the traditional Max-Tree to handle multivariate attributes.

3.1 The Clustering Approach

Here we follow a different approach. Ideally, we would like to select attributes in such a way that vectors belonging to different categories of objects occupy compact and disjoint regions in D-dimensional attribute

vector space. Thus, given a suitable set of attributes or features, we could automatically organize the huge number of connected components of all threshold sets into a much smaller number of groups by automatic clustering. Instead of painstakingly setting reference shape and correct distance threshold, the user now inspects a limited number of clusters.

Let $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$, be set of connected components of image X where $\vec{\tau}(C_i) \in \mathbb{R}^D$ denotes the associated attribute vector. As in [34] $\vec{\tau}$ is the vector attribute function.

Any clustering partitions $\mathbb{R}^D$ into k sets. We denote the partition classes $P_j \subset \mathbb{R}^D$, $j = 1, 2, \ldots, k$. Every vector $\vec{x} \in \mathbb{R}^D$ lies in one or more partition classes. The cluster criterion Λ_j becomes

$$\Lambda_j(C) = (\vec{\tau}(C) \in P_j) \tag{19}$$

i.e. it returns true if the attribute vector of C lies in partition P_j. Replacing the usual criterion (13) in attribute filters with (19), we can draw up the *attribute cluster filter* ψ_{Λ_j}

$$\psi_{\Lambda_j}(X) = \bigcup_{x \in X} \psi^{\Lambda_j}(\Gamma_x(X)). \tag{20}$$

It is trivial to show that ψ_{Λ_j} adheres to all the properties of vector-attribute filters.

Let us look into clustering in a bit more detail. In particular, we can distinguish crisp and fuzzy clustering.

Definition 2. *Ψ_{Λ_j} defined according to* (20) *is a crisp attribute cluster filter if:*

1. $P_j \neq \emptyset,\ \forall j \in \{1, 2, \ldots, k\}$
2. $\cup_{j=1}^{k} P_j = \mathbb{R}^D$
3. $P_j \cap P_i = \emptyset,\ i, j \in \{1, 2, \ldots, k\}$ *and* $i \neq j$

i.e. $\mathbf{P} = \{P_j, j \in \{1, 2, \ldots, k\}\}$, *is a partition.*

Furthermore, we can define membership functions, by which a partition P_j can be conveniently represented by the *partition matrix* $\mathbf{U} = [u_{ij}]_{k \times N}$. The ith row of this matrix contains values of the membership function u_i of the ith partition class P_i of $\mathcal{C}$. It follows from Definition 2 that the elements of P must satisfy the following conditions:

1. $u_{ij} \in \{0, 1\}, 1 \leq i \leq k, 1 \leq j \leq N$
2. $\sum_{i=1}^{C} u_{ij} = 1, \forall j$
3. $\sum_{j=1}^{N} u_{ij} < N, \forall i$

This simply means every component must be assigned to exactly one cluster.

Fuzzy partitions are generalizations which enable us to allow a connected component to belong to more than one class, by defining a degree of membership between 0 and 1 inclusive. This would lead to *fuzzy attribute cluster filters*. This could be particularly useful when the boundaries among clusters are not well separated and ambiguous.

Definition 3. *Fuzzy Partition Matrix* $\mathbf{U} = [u_{ij}]_{k \times N}$ *defines a fuzzy partition if*

1. $u_{ij} \in [0, 1],\ \ \forall i, j$
2. $\sum_{i=1}^{k} u_{ij} = 1,\ \ \forall j$
3. $\sum_{j=1}^{N} u_{ij} > 0,\ \ \forall i$

The first condition states that the membership is bounded between 0 and 1, the second states that the sum of memberships of any single entity over all fuzzy partition class is unity, and the third states that the sum of memberships of all items over any single fuzzy partition classes is non-zero (i.e. no fuzzy partition class is empty). It can readily be verified that crisp partitions are a special case of fuzzy partitions.

Using this notion we can generalize our crisp attribute cluster filter to a fuzzy attribute cluster filter $\psi_{\mathbf{U},j,\epsilon}$ by using criterion

$$\Lambda_{\mathbf{U},j,\epsilon}(C_i) \equiv u_{ij} \geq \epsilon \tag{21}$$

where ϵ defines a membership threshold. If **U** defines a crisp partition, this formulation is equivalent to the crisp definition for any $\epsilon \in (0, 1)$.

4 Attribute Cluster Filter Computation and Implementation

In this section we describe vector-attribute filtering pattern classifications in brief detail.

1. **Feature selection**: A large number of both size and shape attributes for filtering in 3D is now available [7, 8, 38, 41]. These attributes enhance the ability of connected filters to select structures of interest for different imaging modalities. Ideally in selecting the attributes we require these attributes to distinguish patterns belonging to different clusters and be less immune to noise. Currently this is done manually. For efficient computation of the attributes, we utilize the Max-Tree [24]. The Max-Tree is a data structure that was designed for morphological attribute filtering. The filtering process is separated into four stages: build, compute attributes, filter and restitution. It is this filtering process that we change in our research, rather than decision being based on attribute signature of the connected component. The decision is based on the feature vector and class of the component determines whether to be removed or retained.
2. **Clustering**: Clustering is ubiquitous and a wealth of clustering algorithms have been developed to solve different problems in various fields. There is no clustering algorithm that can be universally used to solve all problems [10]. In this research we explore four well researched clustering algorithms: k-means [14], fuzzy c-means (FCM) [1], Vector quantization[12] and Mean Shift [3, 6]. In the final results, we eliminated Vector quantization because it's performance was very similar to k-means but slower.

4.0.1 Size based attributes

For size based attributes we consider volume, surface area, *X-extent*, *Y-extent* and *Z-extent* [38]. Volume is easily estimated by counting the number of voxels that constitute an object. A number of surface area estimates of 3D objects exist in the literature [8, 11, 30]. A straightforward and simple way to obtain a surface area estimate of a 3D object is to count the number of foreground voxels with a surface neighbour in the background as in [19]. A more accurate surface area estimate is obtained through approximating the boundary of a triangular representation, using the marching cubes algorithm [13, 30]. *X-extent*, *Y-extent* and *Z-extent* are computed from the minimum and maximum x, y, and z coordinates values of pixels within each peak component. The extent is computed from the differences (plus one). Unlike the others, these size attributes are not rotation invariant, but if the posture in the scanner is known, they can be useful in combination with prior anatomical knowledge.

4.0.2 Shape based attributes

We considered moment based attributes like non-compactness, elongation, flatness, sparseness, radial moments [7, 38] or non-moment-based measures like sphericity [8]. Briefly here is how these attributes are computed:
The moment-of-inertia $I(C)$of an object can be defined as its tensor which is equivalent to the covariance matrix multiplied by the number of voxels in a connected component(C). The non-compactness attribute $N(C)$ is defined as

$$N(C) = \frac{\mathrm{tr}(I(C))}{V^{\frac{5}{3}}(C)} \tag{22}$$

where $I(C)$ is the inertia tensor, tr denotes the trace, and V is the volume of each connected component. Other moment-invariants which have a straightforward geometric interpretation from the eigenvalues of tensor matrix can be derived. Let $e_1(C)$, $e_2(C)$ and $e_3(C)$ be the three (real) eigenvalues of $I(C)$ such that:

$$|e_1(C)| \geq |e_2(C)| \geq |e_3(C)| \tag{23}$$

The measure of *elongation* $\xi(C)$ is given by

$$\xi(C) = \frac{|e_1(C)|}{|e_2(C)|} \tag{24}$$

while *flatness* $F(C)$ is given by:

$$F(C) = \frac{|e_2(C)|}{|e_3(C)|} \tag{25}$$

Furthermore, let $d_i(C)$ be the lengths of the principal axes such that

$$d_i(C) = \sqrt{\frac{20|e_i(C)|}{V}} \tag{26}$$

Then *sparseness* S(C) is given by

$$S(C) = \frac{\pi d_1 d_2 d_3}{6V} \tag{27}$$

This is the ratio of the volume of an ellipsoid with the principal axes as computed from the moment of inertia tensor, divided by the measured volume of the component. It is unity for solid ellipsoids, and rapidly grows as the shape becomes hollowed out, curved, or tree-like.
Radial moment attributes [7] are 3D counterparts of the 2D method from [44]. For each connected component, compute the centroid co-ordinates($\overline{x}$, $\overline{y}$ and $\overline{z}$), the volume, equivalent to central moments(μ_{000}) of the nodes and $\mu_{r,2\beta}$, r is radius for any β. The radial moment attribute($\psi_\beta(C)$) is then defined as:

$$\psi_\beta(C) = \frac{3}{(2\beta+3)^{\frac{3}{2\beta+3}}(4\pi)^{\frac{2\beta}{2\beta+3}}} \frac{\mu_{000}(C)}{\mu_{r,2\beta}^{\frac{3}{2\beta+3}}(C)} \tag{28}$$

where

$$\mu_{p,q,r} = \sum_C (x-\overline{x})^p + (y-\overline{y})^q + (z-\overline{z})^r g(x,y,z) \tag{29}$$

The last attribute tested is the sphericity $S(C)$, which is computed using surface area($A(C)$) and volume($V(C)$) of each component as

$$S(C) = \frac{\pi^{\frac{1}{3}}(6V(C))^{\frac{2}{3}}}{A(C)}. \tag{30}$$

4.0.3 Computing the Attributes using the Max Tree

For efficient computation of the attributes, we utilize the Max-Tree [24]. The Max-Tree is a data structure that was designed for morphological attribute filtering. The filtering process is separated into four stages: build, compute attributes, filter and restitution. To build a Max-Tree, a variety of fast algorithms is available [17, 24, 39, 40]. The approach consists of arranging the subsets of an image into a tree starting from the root node that acts as a parent to all subsequent nodes. Each node represents a flat zone L_h where a set of pixels adopt a single gray-level value of the highest node within that subset. The image is thresholded at level h to obtain the thresholded set consisting of peak components, P_h^k, whose gray-level ≥ h (k is node index). C_h^k are the components in P_h^k with gray-level h. An example is shown in Fig. 2. During the Max Tree building phase, auxiliary data used for computing the node attributes at a later stage can be collected. The auxiliary data can be used to compute one or more attributes, that describe certain properties of the peak components represented by those nodes.

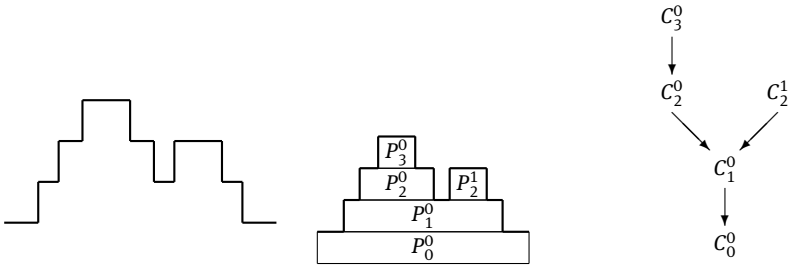

Figure 2: A 1-D signal f (left), the corresponding peak components (middle) and the Max-Tree (right). Figure after [40].

After computing the attributes, the filtering process is based on certain rules like the *Direct*, *Min*, *Max*, and *Viterbi* rules [2, 24], and more recently the *Subtractive* rule [35], and *Branches* rule [16, 22]. Filtering is implemented by checking whether a node, C_h^k, satisfies given criteria in conjunction with the filtering rules. It is this filtering process that we change in our research, rather than decision being based on attribute signature of the connected component, the decision is based on the feature vector and class of the component determines whether to be removed or retained.

4.1 Clustering Methods

4.1.1 K-means

K-means [14] clustering is a method of cluster analysis which aims to partition N observations into k clusters in which each observation belongs to the cluster with the nearest mean. Given a set of objects $x_j \in \Re^d$, $j = 1....N$, the aim is to organize x_j into k subsets $C = \{C_1, ..., C_k\}$. This algorithm aims at minimizing squared error objective function, $J(M)$ defined as

$$J(M) = \sum_{i=1}^{k} \sum_{j=1}^{N} \|x_j - m_i\|^2 \tag{31}$$

where $\|x_j - m_i\|^2$ is a chosen distance measure between a data point x_j and m_i the sample mean for the ith cluster.
The algorithm is as follows:

- Initialize a k-partition randomly or based on prior knowledge
- Calculate the cluster prototype or centroid matrix $M = [m_1, ...m_k]$
- Assign each object in the data set to the nearest cluster C_w
  $x_j \in C_w$, if $|x_j - m_w\| < |x_j - m_i\|$
  for $j = 1, ...N$, $i \neq w$, and $i = 1, ...k$
- Recalculate the cluster prototype matrix based on the current partition
- Repeat steps 2-3 until there is no change for each cluster.

The main advantage of K-means is that it is simple and fast($O(Nkd)$) which allows it to run on large datasets.

4.1.2 Fuzzy C-Means(Fuzzy)

Fuzzy C-means (FCM) [1] is fuzzy clustering method, which allows one piece of data to belong to two or more clusters.
It is based on the concept of fuzzy c-partition [4] summarized previously in Definition 3. To briefly recall this, suppose $X = \{x_1, \ldots, x_N\}$ are a set of a given data where each data point $x_k (k = 1, \ldots, N)$ is a

vector in $\mathbb{R}^D$. The fuzzy cluster memberships are then represented by a real $m \times N$ matrix $\mathbf{U} = [u_{ik}]$, with k be an integer, such that $2 \leq k < N$. Essential constraints are given by

$$\Sigma_{i=1}^{m} u_{ik} = 1 \quad \wedge \quad 0 < \Sigma_{k=1}^{N} u_{ik} < N \tag{32}$$

where u_{ik} is the membership value of x_k in cluster i.
The aim of the FCM algorithm is to find an optimal fuzzy c-partition and corresponding prototypes minimizing the objective function $J_w(U, C; X)$

$$J_w(U, C; X) = \sum_{k=1}^{N} \sum_{i=1}^{m} (u_{ik})^w \|x_k - c_i\|^2 \tag{33}$$

where $C = (c_1, \ldots c_m)$ is a matrix of unknown cluster centers $c_i \in \Re^D$, $\|x_k - c_i\|^2$ is the squared Euclidean distance between a data point x_j and the cluster centre c_i and $w \in [1, \infty)$ is a constant that influences the membership values. To minimize criterion J_w, under the fuzzy constraints defined in eqn. 32, fuzzy partitioning is carried out through an iterative optimization of the objective function J_w, with the update of membership u_{ik} and the cluster centers c_i by:

$$u_{ik} = \left[\sum_{j=1}^{m} \left(\frac{\|x_k - c_i\|^2}{\|x_k - c_j\|^2} \right)^{\frac{2}{w-1}} \right]^{-1} \tag{34}$$

$$c_i = \frac{\sum_{i=1}^{m} u_{ik}^w x_i}{\sum_{i=1}^{m} u_{ik}^w} \tag{35}$$

This iteration will stop when:

$$\max_{ik} |u_{ik}^{j+1} - u_{ik}^{j}| < \epsilon$$

where $\epsilon \in [0, 1]$, whereas j are the iteration steps.
The algorithm is as follows:

- Initialize the number of clusters.
- Set ϵ and cluster membership threshold.
- Initialize $U = [u_{ik}]$ matrix, for the values of the memberships of points.
- At each k-step: calculate the centers vectors using eqn. 35
- Update the matrix U using eqn. 34 and membership.
- Iterate until $\{u_{ik}^{j+1} - u_{ik}^{j}\} < \epsilon$.

4.1.3 Mean Shift Clustering

The k-means and fuzzy c-means algorithms require the number of clusters to be known beforehand. This requires some knowledge about the structure of the data set. However, with increased dimensionality and number of data points in the order of 10^6, it becomes difficult to manually inspect the data points to determine the number of clusters. The mean-shift algorithm [6] computes the number of clusters automatically.
The algorithm takes the following steps:

- Determine features upon which clustering is required and in our case compute connected components attributes.
- Initialize scale or bandwidth at individual feature points
- For each point x_j:
  - Choose a search window
  - Find the mean of all points in the search window centred on the data point and compute the mean shift vector $m_h(x_j)$

- Repeat by searching the window centred on the mean from the previous step.
- Stop when successive means are the same or the shift is less than a threshold. This mean is the value of the peak.

– Assigned all data points with the same peak to the same cluster.

It must be noted that the mean shift algorithm in its original form is too slow especially for large data points in the region of 10^5 and high dimensions(≥ 3). To speed it up a number of methods have been proposed, like nearest neighbor lookups or triangle inequalities [3, 5, 37]. The simplest being upon finding a peak, each data point that is at a distance $\leq t$ from the peak with the cluster is defined by that peak. This is based on the intuition that points that are within one window size distance from the peak will, with high probability, converge to that peak. This is sometimes referred to as basin of attraction. It's this speedup that we utilize in our implementation.
The advantage of using this algorithm is that there is no need to guess the number of clusters. Besides, bandwidth parameter gives some degree of control, and methods of bandwidth estimation have been published [29]. A disadvantage is that the algorithm is very slow ($O(N^2)$). Besides, there is a tendency to get small clusters in regions of low density. The latter is visible in our application to medical image filtering. Furthermore, the output depends on bandwidth and there is ambiguity in optimal selection of bandwidth h. Finally, it does not scale well with increased dimension of feature space.

3. **Implementation**: We implemented vector-attributes for 3D grey-scale attribute filtering in the MTdemo package [38]. This uses the Max-Tree [24] data structure to compute and visualize volumetric data. In [38], Max-Tree construction and attribute computation are separated, allowing computation of different attributes without complete re-building of the tree. However, filtering is based on a single attribute property in this structure, and the notion of filtering is hard-coded as comparison to a threshold λ. To accommodate vector-attribute computation and attribute cluster filtering, a number of changes are included in the structure: more notably, *NodeSelector* abstract class, which enables us to use any type of filtering criteria other than attribute signature; *clusterID* – an extra field per node is required – this field per node is required, this field stores the cluster label of the node, *ComputeAttributes()* function used in the vector-attributes computation class to compute any number of attributes and store the attributes for each node in an auxiliary array index or vector.

```
procedure ComputeAttributes(MaxTree mt, void * attr, num)
    build Max-Tree for the image
    var idx
    var *node
    var *vec
for i = 1 → num do
    attr[i] → calculateAttributes(mt)
 forall the nodes in Max-Tree do
        compute node index idx
        find node's attribute
        vec[idx]= node[idx] → attribute
     end
    end
end procedure
```

Algorithm 1: The vector-attribute computation Algorithm

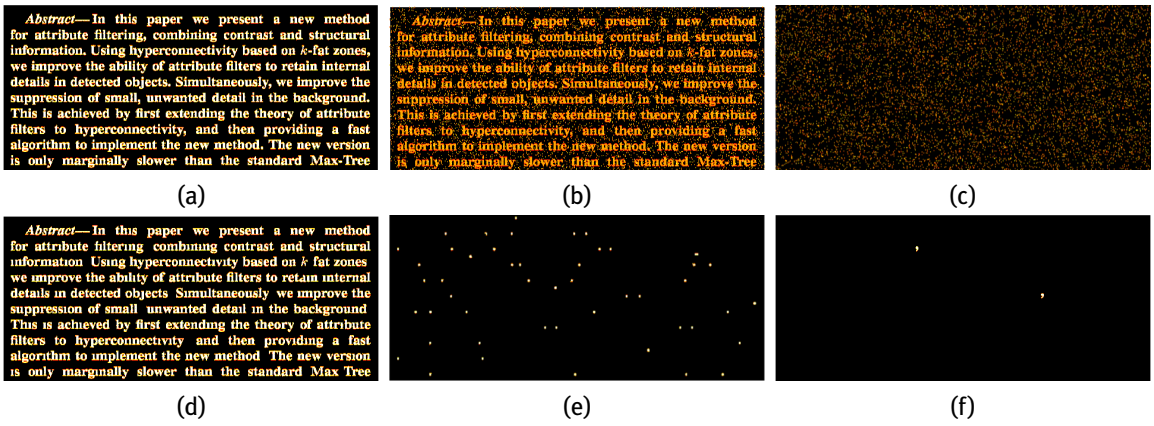

Figure 3: Mean shift attribute cluster filtering of document: (a) Original (b) corrupted (c,d,e,f) the different clusters

5 Results and Discussion

To demonstrate the performance of attribute-cluster filtering, we ran tests on a number of 3D medical images of different modalities, courtesy of http://www.volvis.org and the Department of Radiology and Medical Imaging, University General Hospital of Alexandroupolis, Greece. In the following, we often use just a single attribute, in order to compare more reasonably with classical attribute filters based on attribute thresholds. Due to difficulties in quantifying filtering results in these cases, we first perform a more quantitative test on a printed document.

Document Validation: We begin this section by demonstrating how attribute cluster filters are computed through a simple example of a document that has been corrupted with salt and pepper noise of density 0.3 as shown in Fig. 3. We then apply an attribute cluster filter using the volume attribute (equivalent to area filtering in 2-D) to the document which yields four foreground clusters: (i) the cluster containing the alphabet characters, (ii) the noise, (iii and iv) the punctuation marks. A close inspection shows that this is a valid decomposition of the original document. All clusters are shown in Fig. 3. To provide a quantitative measure of the quality of these methods in image filtering we used the document and filtered the document using the area attribute. Performance analysis of attribute-cluster filtering is carried out using universal quality index (UQI), which models image distortion as a combination of loss of correlation, luminance distortion, and contrast distortion. The regular attribute filter using a manually selected area-threshold achieves a UQI index of 0.91 while the attribute-cluster filter achieves a UQI index of 0.89, both with respect to the original, uncorrupted document see Fig. 10. Thus the automatic thresholds chosen by k-means clustering yield a result very close to the manual method.

5.1 MRI and CT Scan Performance

The angiolarge: In this experiment we compared the performance of the attribute cluster filter against the conventional attribute filters. In their current format, attribute filtering for 3d medical images based on size-based attributes perform very poorly, they not only fail in enhancing blood vessels but also amplify noise. This is illustrated in Fig. 4 (b), a *volume* attribute ($\lambda = 9000$) simply amplifies noise on this data set. This also applies for all size based attributes like *surface area, X-extent, Y-extent, Z-extent*. However, when attribute cluster filter is applied on the *volume* attribute, the performance of this attribute is seen in Fig. 4(c). The noise is not only suppressed but the blood vessels are clearly enhanced. In this case the k-means clustering was used with the number of clusters $k = 11$. The result presents one of the 11 clusters computed for the data set but this was selected as a basis for comparison because in this particular cluster more blood vessels were retained. The other clusters can be availed on request. The performance of the other size based attributes is

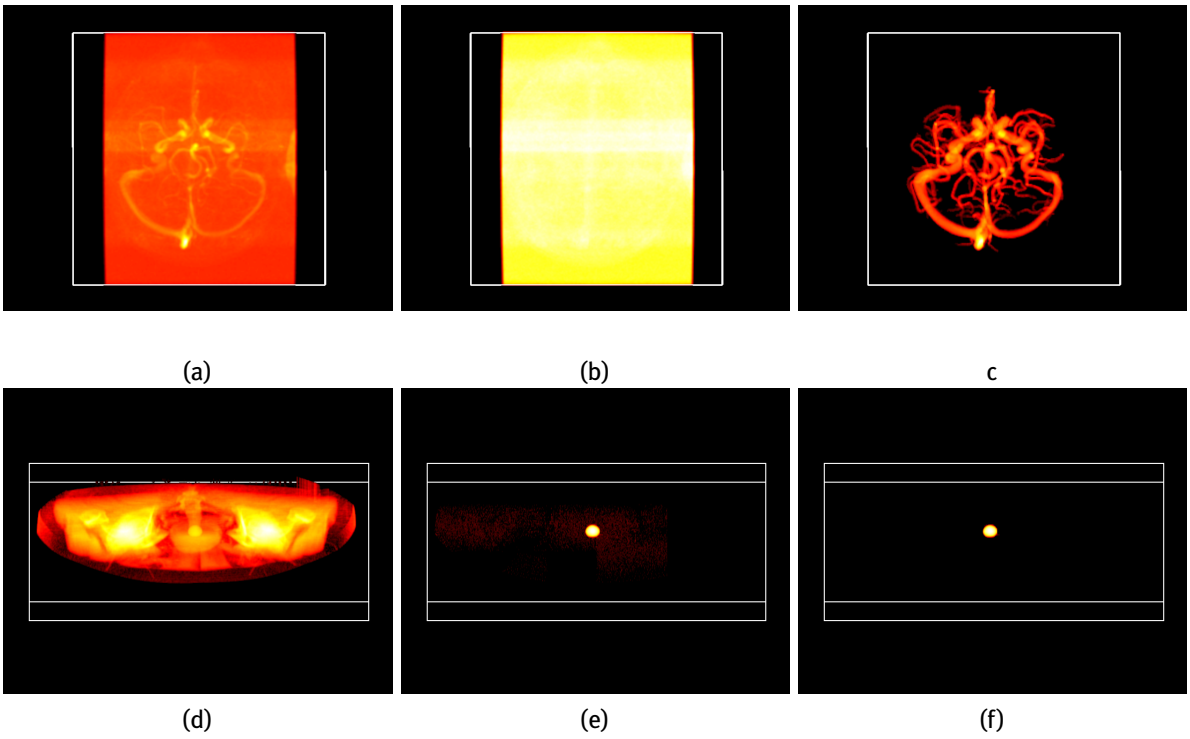

Figure 4: (a) original; (b) filtered with volume($\lambda = 9000$); (c) attribute cluster filter by the same attribute using k-means with $k = 8$; (d) original; (e) filtered with radial moment ($\beta = 5, \lambda = 0.00256$); (f) attribute cluster filter by the same attribute using k-means with $k = 17$

also improved by using attribute cluster filtering. It's important to note that the performance improvement is irrespective of the increase in the dimensionality of the vector-attributes used. We tested this up to 6 attributes. The other clustering techniques are also able to achieve this result but at a higher computational time. Shape based attributes like *non-compactness, radial moment* always perform well on this data set even with simple attribute thresholding.

The prostate-stone: On this data set, attribute filters in their current format are able to isolate the stone but they are never successful in suppressing the noise Fig. 4(e). The problem has been eradicated in [7] [8] by filtering using 2 attributes successively. First, a *non-compactness, or sphericity or radial moment* is applied to the data set to obtain result shown in Fig. 4(e), then a *volume or surface area* is applied to remove the remaining noise. However, using attribute cluster filtering, the result in Fig. 4(f) is obtained in a single step. This result was obtained using k-means and the *non-compactness* attribute with $k = 17$. Higher dimension of the vector up to 5 attributes was capable of isolating the stone in a single step but at a higher computational cost. The other clustering techniques are also able to achieve this result with mean shift clustering using 3 attributes, while fuzzy c-means uses a single attribute but at a higher computational cost, as compared to k-means.

5.2 CT-Knee Volume

The CT-Knee is 8 bit, 379 ×229 ×305 volume. The goal here is to correctly enhance the bones but suppress the tissue. Attribute filters in their current format struggle on this data set, all attributes when used failed to suppress the noise in this data set. This is seen in Fig. 5(b), the *volume* attribute completely fails to filter anything while the *non-compactness* tries to enhance the bones Fig. 5 (c), but still it does fail to suppress all the noise. When 2 attributes are used successively the result does not improve either Fig. 5(d). However, the attribute cluster filter not only suppresses the noise but also enhances the bones Fig. 5(e). This is achieved

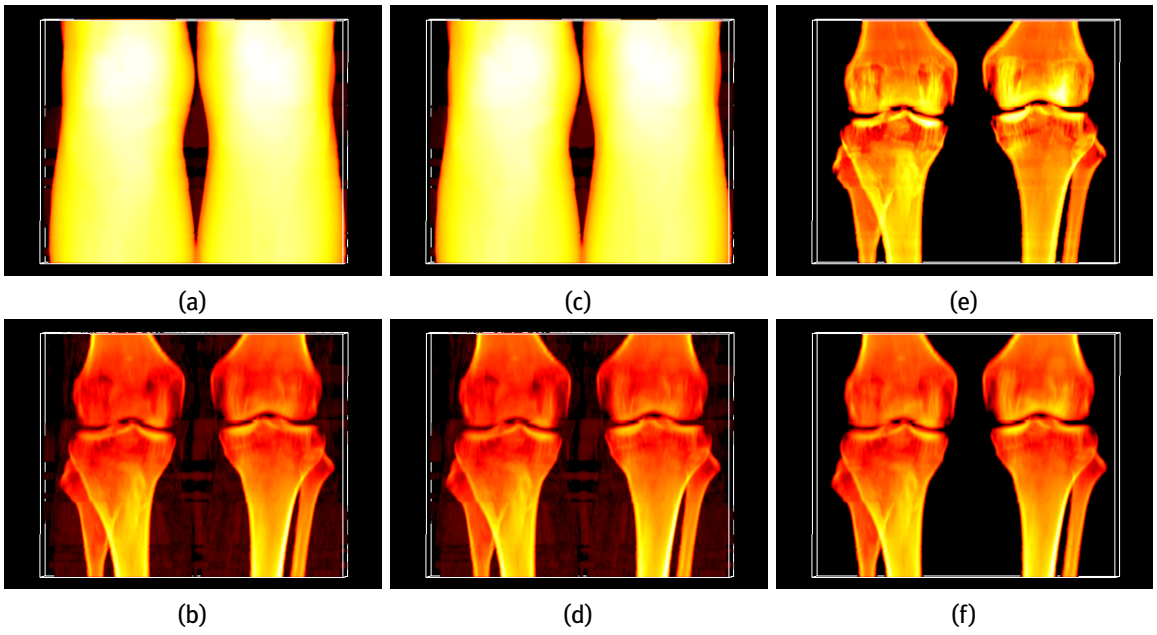

Figure 5: CT-Knee Filtering:(a)original (b)filtered with volume($\lambda = 4000$) (c)filtered with non-compactness($\lambda = 3.5$) (d)successive filter of non-compactness($\lambda = 3.5$) and volume($\lambda = 2000$) (e) attribute cluster filter using 5 attributes and k-means($k = 15$) (e) k-flat filtering($k = 70$)

using both fuzzy c-means and k-means with $k = 15$ using 5 attributes of *surface area, X-extent, Y-extent, Z-extent, volume*. The same result is only matched using hyperconnectivity [20] Fig. 5(f), with hyperconnectivity k-flat zones [20], which are connected regions of maximal extent, where the total grey level variation is not more than k. This restriction to grey-level range automatically restricts the size to which the regions can grow, yielding overlapping pseudo-flat zones, which improves the enhancement of internal details. The effect of using k-flat zones means that any node in the Max-Tree within k of an extremum is not considered an independent entity and their attributes are ignored in any further computation. The mean shift clustering completely decomposes this data set in a very interesting manner which gives more insight into this data set.

5.3 Foot Volume

The human foot: Attribute filters normally struggle to suppress noise on this data set, as seen in Fig. 6. In Fig. 6(b), the *non-compactness* attribute is able to enhance the bones but with noise still visible, while in Fig. 6(c) the *surface area* attribute, like all other size based attributes, simply amplifies noise. However, attribute cluster filter using a combination of *surface area[8], surface area[19] and volume* perfectly enhances the bones and suppresses noise as seen in Fig. 6(d). All the clustering techniques perform well on this data set.

5.4 Kidney-stone

The kidney-stone: The kidney-stone data set is more complex and has poor soft-tissue contrast, low SNR and shading effect. The kidney-stone data set is very hard to filter for most attributes when done on a single attribute. The performance of regular attributes on this data set is shown in Fig. 7(b) for volume attribute and Fig. 7(c) for the radial moment attribute. All regular size based attribute filtering performs poorly in filtering out the kidney stone as seen in Fig. 7(b), while for the shape based attributes only radial moment does a

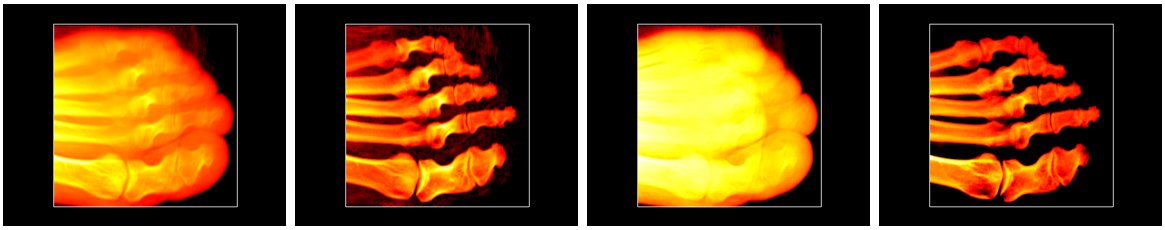

Figure 6: foot filtering: (a) original (b) filtered with non-compactness ($\lambda = 1.3$ (c) filtered with volume($\lambda = 7000$) (d) attribute cluster filter using fuzzy c-means with combination of surface area, volume, surface area $k = 8$

relative good job but still struggles to suppress noise as shown in Fig. 7(c). To suppress the noise the surface area or volume attributes are applied to Fig. 7(c) like in [7]. However, with attribute cluster filtering using mean shift algorithm and 5 size based attributes in a single step the kidney-stone is isolated without any noise as shown in Fig. 7(d). Attribute cluster filtering also reveals that there are more than one stone Fig. 7(f). This attribute filtering method is able to clearly enhance bony like structures, such as part of the spinal cord Fig. 7(e). The k-means and fuzzy c-Means fail to isolate the kidney stone but succeed in revealing the spinal cord.

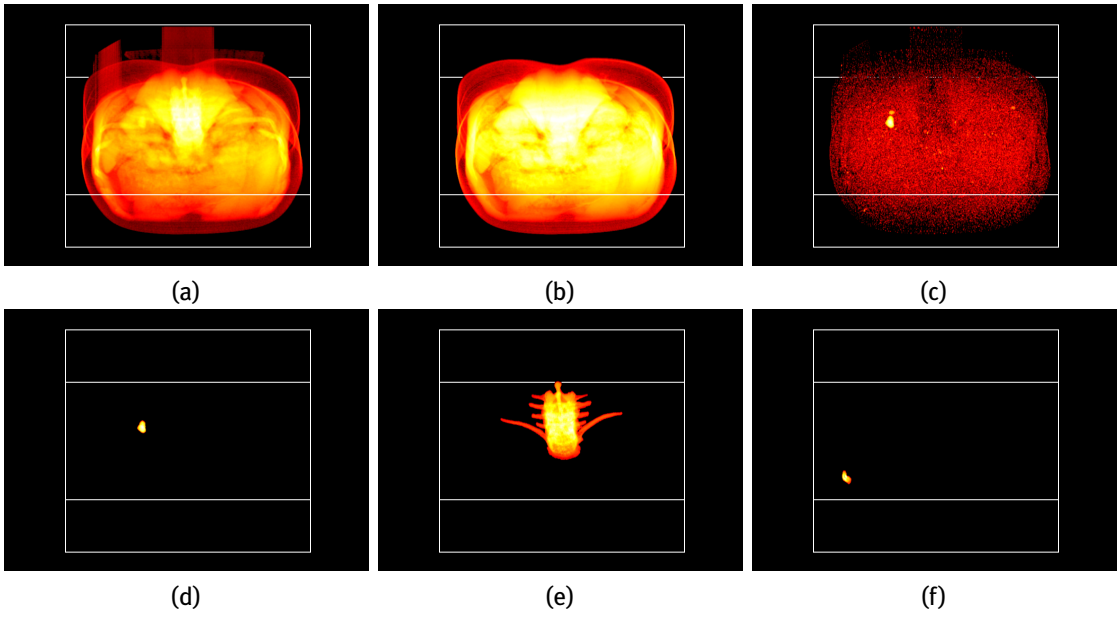

Figure 7: kidney-stone filtering: (a) original; (b) filtered with volume (λ=7400); (c) filtered with radial moment (β=3, λ=0.124); (d,e,f) the result of attribute cluster filter using the mean shift

5.5 Female Chest

The female Chest: From Fig. 8 the performance of regular attribute filter is seen in Fig. 8(e), the radial moment(β) attribute is able to enhance the skeleton but other unwanted tissue still remains. However, an attribute cluster filter of any combination of size attributes not only enhances the skeleton without leaving unwanted tissue but also enhances the heart, though faintly seen, as in Fig. 8(f). All clustering techniques

attain this result. In this experiment volume and non-compactness attributes were used with fuzzy c-means $k = 9$.

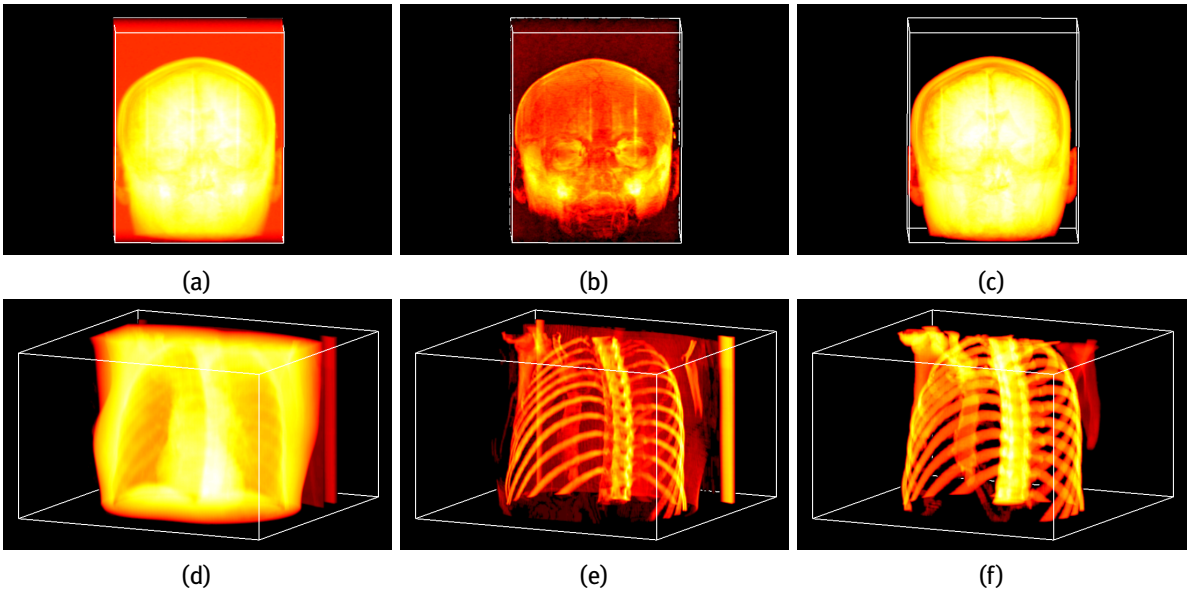

Figure 8: Female-chest/Head Filtering: Left column (a) original head (c) filtered with non-compactness ($\lambda = 2.6$) (e) attribute cluster filter using k-means ($k = 9$); Right column (b) original chest (d) filtered with radial moment ($\beta = 3, \lambda = 0.034$); (f) attribute cluster filter using fuzzy c-means ($k = 12$)

5.6 MRI-Head

The human Head: From Fig. 8 the performance of regular attribute filter is seen in Fig. 8(b), the non-compactness attribute enhances the head but other unwanted tissue still remains. However, an attribute cluster filter combination of any number of size attributes gives the result seen in Fig. 8(c). All noise is suppressed.

Computational Timings: Using a standard Core 2 Duo E8400 at 2.0 GHz, 2GB RAM machine we ran timings for the computation of each algorithm for attribute cluster filtering up to 6 attributes for different medical images of varying sizes and gray scale levels. The timings include the computation of the attributes and the clusters. The number of clusters computed was determined by the mean shift method because in this algorithm the number is not explicitly pre-determined thus we used it as the reference point. We consider the following attributes: *volume, surface area[8], surface area[19], X-extent, Y-extent and Z-extent*, all these attributes are size based. The results of the various clustering methods are shown in Table 1. K-means algorithm is by far the faster algorithm, as expected, others follow interchangeably. The computation of shape based attributes is slower by an average factor of three, this is because of the floating values of shape descriptors. But overall the computational complexity of these methods is good even for large data sets, for instance, even for very large data sets like mrt16_angio with 1,554,454 nodes for 6 attributes for $k = 23$: it takes 17 seconds for size based attributes and 58 seconds for shape attributes. This is faster and most users can select an optimal setting for a single attribute.

Table 1: Cluster Computing Time(seconds)

Dataset	No.Nodes	No.Clusters	K-means	Fuzzy	MSF
angiolarge	361,463	10	3.02	170	172
mrt16_angio	1,554,454	7	12.1	819	525
mrt16_angio2	419,454	11	5.74	389	148
foot	279,513	11	5.50	64.0	62.9
prostate-stone	124,477	9	1.58	18.2	26.0
Kidney-stone	387,462	17	17.6	149	202
CT-Knee	117,920	23	12	71	74
Head	752,333	9	24	168	123
Chest	85,414	15	5.2	30.6	24.7
Document	290,446	3	0.11	0.40	8.20

5.7 Discussion

Various combinations of the attributes were used to evaluate the performance of different attributes in correctly clustering different data sets. The performance of the combination of size based attributes was good on data sets that involved separating hard tissue from soft tissue, that is the CT scans *foot, chest knee, kidney-stone*. This is in part due to the fact that these structures are brighter, and so are the peaks in the Max-Tree. The combination of shape based attributes performed better on enhancing and noise suppression on data sets that exclusively were made of soft tissue, that is the MRI *angiolarge, mrt16_angio, mrt16_angio2* . In clustering, size attributes dominate over shape. On the *angiolarge* data set the *non-compactness* normally does a good job in terms of vessel filtering and noise suppression, but *volume* always performs poorly. Volume has larger values as compared to non-compactness, and therefore volume decides the clustering results and thus the filtering looks more like volume attribute filter, see Fig. 9. This has to do with the use of the Minkowski metric where the largest scaled features dominate others. This could be solved by normalizing the volume attributes but in this research we did not explore this. Experiments also show that with this kind of mixture, more structures seem to be retained when a pre-filter is first applied to very difficult data sets, such as time of flight.

The clustering of scalar attributes (i.e. $d = 1$) using a suitable number of clusters for all attributes and most data sets gives very good results, as compared to regular attribute filtering, irrespective of whether it is a shape or size based attribute. A further increment in the number of attributes to more than 6 reveals little or no changes in performance for both size and shape based attributes. This could be due to the distance used in the clustering process as the similarity measure. Normalization or relevance learning could be used to combine features in a better way. The mean-shift algorithm has particular difficulties, where the performance in high dimensions degrades rapidly, possibly through the sparseness of data space.

Overall, all three clustering algorithms succeed on a number of data sets. By far k-means performs much better than the others while the mean shift looks promising with a number of great results. Perhaps the major concern of these clustering techniques is how to determine the *optimal* number of clusters. The mean shift determines this automatically but it has so many parameters that need to be set.

6 Conclusion

In this paper, we presented methods for computing attribute-cluster filtering in 3-D using unsupervised pattern classification where image or volume features are selected or rejected based on feature vectors rather than a single property. We have shown that the performance of attribute-cluster filters is better than those of regular attribute filters in enhancing structures in medical images and noise suppression in most cases.

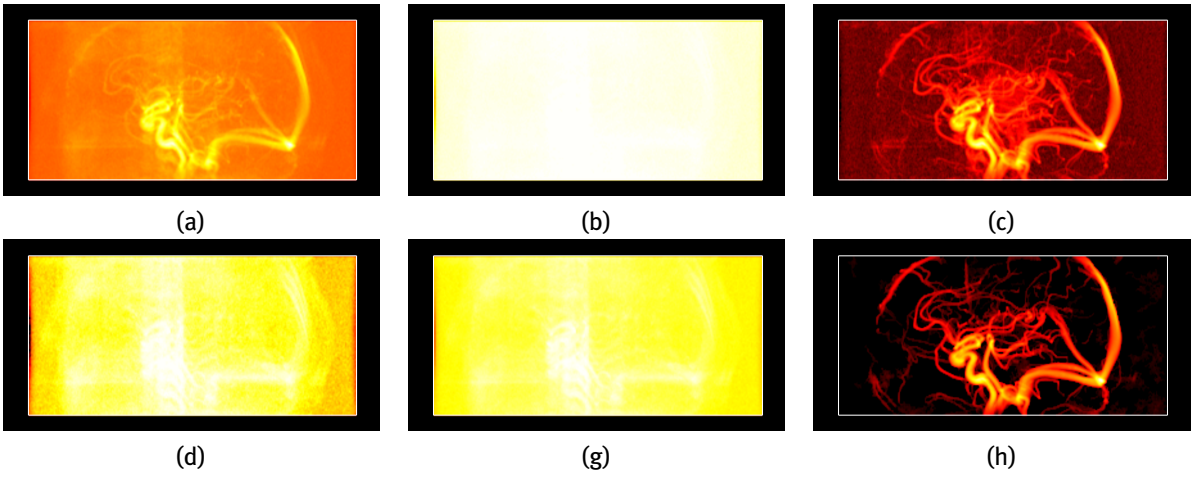

Figure 9: Attribute cluster filter using a combination of volume and non-compactness: (a) original (b,c,d) clusters returned using k-means; (g) volume (λ=1500) (h) non-compactness (λ=2.7)

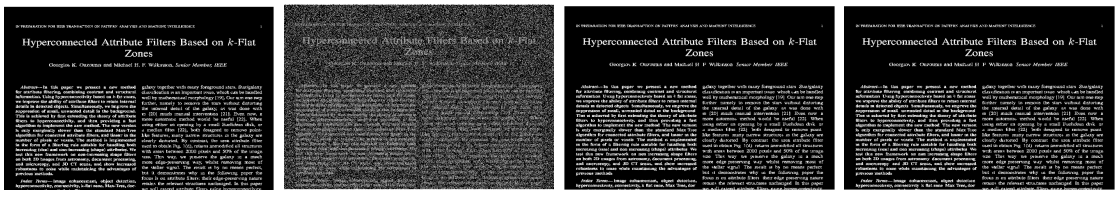

Figure 10: Document filter validation: (a) original; (b) corrupted; (c) filtered with area ($\lambda = 65$); (d) vector-attribute using k-means($k = 4$)

These filters show a lot of flexibility in selecting features of interest. Though the implementation of these techniques is not very sophisticated, their computational load is already acceptable. Algorithmic advances could improve this further. Attribute cluster filtering using k-means is fastest and gives good results. The fuzzy c-means is slower but allows us to have flexibility in deciding cluster membership, especially in images without clear boundaries, which are prevalent in medical images. The mean-shift method though slow as expected, performs well using standard kernel estimation profiles. Through its automatic cluster learning and unique image decomposition it exhibits a lot of potential for further investigation.

In future work, we will study the behavior of attribute cluster filters for higher dimensional (≥ 10) attribute vectors. The current metric used in the clustering process as the similarity measure is not suitable in high dimensional space and certainly needs rescaling to better combine attributes with very different ranges. Thus, we intend to look at other metrics. Dimensionality reduction techniques are also required, in part to reduce the actual clustering times, but also because it is clear that not all the attributes contribute equally to the separation of the data.

References

[1] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA (1981)

[2] Breen, E.J., Jones, R.: Attribute openings, thinnings and granulometries. Comp. Vis. Image Understand. 64(3), 377–389 (1996)

[3] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24(5), 603–619 (May 2002)

[4] Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Journal of Cybernetics 3(3), 32–57 (1973)
[5] Freedman, D., Kisilev, P.: Fast mean shift by compact density representation. In: CVPR. pp. 1818–1825. IEEE (2009)
[6] Fukunaga, K., Hostetler, L.D.: Estimation of the gradient of a density function with applications in pattern recognition. IEEE Trans. Inform. Theor. IT-21, 32–40 (1975)
[7] Kiwanuka, F.N., Wilkinson, M.H.F.: Radial moment invariants for attribute filtering in 3D. In: Käthe, U., Montanvert, A., Soille, P. (eds.) Proc. Workshop on Applications of Discrete Geometry and Mathematical Morphology (WADGMM). pp. 37–41 (2010)
[8] Kiwanuka, F.N., Ouzounis, G.K., Wilkinson, M.H.: Surface-area-based attribute filtering in 3d. In: Proceedings of the 9th ISMM '09. pp. 70–81. Springer-Verlag, Berlin, Heidelberg (2009)
[9] Kiwanuka, F.N., Wilkinson, M.H.F.: Cluster-based vector-attribute filtering for ct and mri enhancement. In: ICPR. pp. 3112–3115. IEEE (2012)
[10] Kleinberg, J.: An impossibility theorem for clustering. pp. 446–453. MIT Press (2002)
[11] Lindblad, J.: Surface area estimation of digitized 3d objects using weighted local configurations. Image and Vision Computing 23(2), 111 – 122 (2005)
[12] Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. Communications, IEEE Transactions on 28(1), 84–95 (jan 1990)
[13] Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. Computer Graphics 21(4), 163–169 (1987)
[14] Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297 (1967)
[15] Maragos, P., Ziff, R.D.: Threshold decomposition in morphological image analysis. IEEE Trans. Pattern Anal. Mach. Intell. 12(5), 498–504 (1990)
[16] Naegel, B., Passat, N., Boch, N., Kocher, M.: Segmentation using vector-attribute filters: Methodology and application to dermatological imaging. In: Proc. Int. Symp. Math. Morphology (ISMM) 2007. pp. 239–250 (2007)
[17] Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Proc. 15, 3531–3539 (2006)
[18] Ouzounis, G.K., Giannakopoulos, S., Simopoulos, C.E., Wilkinson, M.H.F.: Robust extraction of urinary stones from ct data using attribute filters. In: Image Processing (ICIP), 2009 16th IEEE International Conference on. pp. 2629 –2632 (nov 2009)
[19] Ouzounis, G.K., Giannakopoulos, S., Simopoulos, C.E., Wilkinson, M.H.F.: Robust extraction of urinary stones from ct data using attribute filters. In: Proc. Int. Conf. Image Proc. 2009 (2009)
[20] Ouzounis, G.K., Wilkinson, M.H.F.: Hyperconnected attribute filters based on k-flat zonesg. IEEE Trans. Pattern Anal. Mach. Intell. (2010), in press
[21] Perret, B., Collet, C.: Connected image processing with multivariate attributes: An unsupervised markovian classification approach. Computer Vision and Image Understanding 133(0), 1–14 (2015)
[22] Purnama, K.E., Wilkinson, M.H.F., Veldhuizen, A.G., van Ooijen, P.M.A., Lubbers, J., Sardjono, T.A., Verkerke, G.J.: Branches filtering approach for max-tree. In: Proc. 2nd Int. Conf. Comput. Vision Theory Applic (VISAPP) 2007. pp. 328–332. Barcelona (March 8-11 2007)
[23] Ronse, C.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. J. Math. Imag. Vis. 8, 41–58 (1998)
[24] Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7, 555–570 (1998)
[25] Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Trans. Image Proc. 4, 1153–1160 (1995)
[26] Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing 4, 1153–1160 (1995)
[27] Salembier, P., Wilkinson, M.H.F.: Connected operators: A review of region-based morphological image processing techniques. IEEE Signal Processing Magazine 26(6), 136–157 (2009)
[28] Serra, J.: Connectivity on complete lattices. J. Math. Imag. Vis. 9(3), 231–251 (1998)
[29] Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
[30] Sladoje, N., Nystrom, I., Saha, P.K.: Measurements of digitized objects with fuzzy borders in 2d and 3d. Image Vis. Comput. 23, 123 – 132 (2005)
[31] Soille, P.: Constrained connectivity and connected filters. IEEE Trans. Pattern Anal. Mach. Intell. 30(7), 1132–1145 (July 2008)
[32] Tushabe, F.: Extending Attribute Filters to Color Processing and Multi-Media Applications. Ph.D. thesis, University of Groningen (2010)
[33] Tushabe, F., Wilkinson, M.H.F.: Image preprocessing for compression: Attribute filtering. In: World Congress on Engineering and Computer Science 2007 (2007)
[34] Urbach, E.R., Boersma, N.J., Wilkinson, M.H.F.: Vector-attribute filters. In: Mathematical Morphology: 40 Years On, Proc. Int. Symp. Math. Morphology (ISMM) 2005. pp. 95–104. Paris (18-20 April 2005)
[35] Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Trans. Pattern Anal. Mach. Intell. 29, 272–285 (2007)

[36] Urbach, E.: Contextual image filtering. Image and Vision Computing New Zealand, 2009. IVCNZ '09. 24th International Conference pp. 299 –303 (nov 2009)
[37] Wang, P., Lee, D., Gray, A.G., Rehg, J.M.: Fast mean shift with accurate and stable convergence. Journal of Machine Learning Research - Proceedings Track 2, 604–611 (2007)
[38] Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the max-tree representation. IEEE Trans. Image Proc. 16, 2943–2952 (2007)
[39] Wilkinson, M.H.F.: A fast component-tree algorithm for high dynamic-range images and second generation connectivity. In: Proc. Int. Conf. Image Proc. 2011. pp. 1041–1044 (2011)
[40] Wilkinson, M.H.F., Gao, H., Hesselink, W.H., Jonker, J.E., Meijster, A.: Concurrent computation of attribute filters using shared memory parallel machines. IEEE Trans. Pattern Anal. Mach. Intell. 30(10), 1800–1813 (2008)
[41] Wilkinson, M.H.F., Westenberg, M.A.: Shape preserving filament enhancement filtering. In: Niessen, W.J., Viergever, M.A. (eds.) Proc. MICCAI'2001. Lecture Notes in Computer Science, vol. 2208, pp. 770–777 (2001)
[42] Wilkinson, M., Ouzounis, G.K., Soille, P.: Concurrent computation of differential morphological profiles for remote sensing. In: Käthe, U., Montanvert, A., Soille, P. (eds.) Proc. Workshop on Applications of Discrete Geometry and Mathematical Morphology (WADGMM). pp. 67–71 (August 2010)
[43] Xu, Y., Géraud, T., Najman, L.: Morphological filtering in shape spaces: Applications using tree-based image representations. CoRR abs/1204.4758 (2012)
[44] Zunić, J., Hirota, K., Rosin, P.L.: A Hu moment invariant as a shape circularity measure. Pattern Recogn. 43(1), 47–57 (2010)